Data Quality Report

Table of Contents

Introduction to the dataset

In order to get a stronger perspective on our data, there are some basic steps we check first. For example, what does our data look like / how big is it? Knowing how many rows and columns we have allows us to compare our cleaned version of the data against our original dataset. So if we start with 1000 and end up with 900, it provides some information on how much data was unusable in the original data set.

1.1 Datatypes

The first thing I noticed about the data was that all datatypes, except one where RiskPerformance was graded as “Good” or “Bad”, were listed as integers. Intuitively, this may not reflect the data in a manner which was useful, and it was worth noting for further inspection.

In order to do some of our plotting, we needed to decide how to classify each column. Category, float and integer datatypes were chosen for various columns in order to allow us to plot before doing our data quality plan.

1.2 Data missing a unique identifier

Another thing to note was that the rows in the data have no unique identifier. This causes some difficulty when exporting the updated data frame with pandas.

1.3 Duplicate data

Another useful piece of information is to identify how much duplicate data we have. These duplicate data are most likely incorrect and might provide heavier weighting to portions of the data. This means that certain points in the data will be overrepresented.

1.4 Constant Columns

No constant columns were present in the dataset.

1.5 Null values

When it comes to doing descriptive statistics, an issue was highlighted where some rows were not being considered. In order to identify why, I first checked whether there was any value in these rows – and if not, it would make sense that they had been excluded. There were 49 rows that had null values for 3 separate columns.

1.6 Continuous Plots

1.6.0 Statistical Overview

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	949.0	498.696523	290.545759	0.0	246.0	501.0	750.0	999.0
ExternalRiskEstimate	949.0	71.800843	10.362591	-9.0	65.0	72.0	80.0	93.0
MSinceOldestTradeOpen	949.0	194.305585	99.624429	-9.0	130.0	177.0	261.0	530.0
MSinceMostRecentTradeOpen	949.0	9.374078	11.781243	-9.0	3.0	6.0	11.0	156.0
AverageMInFile	949.0	77.778714	32.664086	-9.0	58.0	75.0	95.0	257.0
NumSatisfactoryTrades	949.0	20.581665	11.178024	-9.0	12.0	19.0	27.0	65.0
NumTrades60Ever2DerogPubRec	949.0	0.679663	1.733456	-9.0	0.0	0.0	1.0	19.0
NumTrades90Ever2DerogPubRec	949.0	0.453109	1.527236	-9.0	0.0	0.0	0.0	19.0
PercentTradesNeverDelq	949.0	91.970495	12.489285	-9.0	88.0	97.0	100.0	100.0
MSinceMostRecentDelq	949.0	7.759747	20.299062	-9.0	-7.0	1.0	16.0	82.0
NumTotalTrades	949.0	21.626976	12.232796	-9.0	13.0	20.0	29.0	68.0
NumTradesOpeninLast12M	949.0	1.891465	1.943240	-9.0	0.0	1.0	3.0	12.0
PercentInstallTrades	949.0	35.463646	17.344852	-9.0	23.0	33.0	45.0	100.0
MSinceMostRecentInqexcl7days	949.0	0.142255	5.742340	-9.0	0.0	0.0	1.0	23.0
NumInqLast6M	949.0	1.420443	2.017655	-9.0	0.0	1.0	2.0	21.0
NumInqLast6Mexcl7days	949.0	1.353003	1.978456	-9.0	0.0	1.0	2.0	21.0
NetFractionRevolvingBurden	949.0	34.987355	29.802866	-9.0	8.0	31.0	57.0	165.0
NetFractionInstallBurden	949.0	41.799789	42.105152	-9.0	-8.0	50.0	82.0	140.0
NumRevolvingTradesWBalance	949.0	3.732350	3.306115	-9.0	2.0	3.0	5.0	23.0
NumInstallTradesWBalance	949.0	1.604847	3.374766	-9.0	1.0	2.0	3.0	19.0
NumBank2NatlTradesWHighUtilization	949.0	0.513172	2.674912	-9.0	0.0	1.0	1.0	16.0
PercentTradesWBalance	949.0	66.678609	22.880237	-9.0	50.0	67.0	83.0	100.0

```
Unnamed: 0                          501.0
ExternalRiskEstimate                 72.0
MSinceOldestTradeOpen               177.0
MSinceMostRecentTradeOpen             6.0
AverageMInFile                       75.0
NumSatisfactoryTrades                19.0
NumTrades60Ever2DerogPubRec           0.0
NumTrades90Ever2DerogPubRec           0.0
PercentTradesNeverDelq               97.0
MSinceMostRecentDelq                  1.0
MaxDelq2PublicRecLast12M              6.0
MaxDelqEver                           6.0
NumTotalTrades                       20.0
NumTradesOpeninLast12M                1.0
PercentInstallTrades                 33.0
MSinceMostRecentInqexcl7days          0.0
NumInqLast6M                          1.0
NumInqLast6Mexcl7days                 1.0
NetFractionRevolvingBurden           31.0
NetFractionInstallBurden             50.0
NumRevolvingTradesWBalance            3.0
NumInstallTradesWBalance              2.0
NumBank2NatlTradesWHighUtilization    1.0
PercentTradesWBalance                67.0
```

Figure 1 Median Values for Continuous Features

The first thing that draws my attention is our minimum column. All of the values here are -9.0. While this 9.0 may hold significant meaning as per the accompanying documentation with the dataset, this unfortunately skews our ability to investigate the true minimum for each of these respective columns. This issue should be considered for the data quality plan.

The second is that row IDs are currently represented as “unnamed” in our data frame and should be corrected prior to the quality plan.

Next, I’m looking at the discrepancy between any mean/median values to discern if there are issues with outliers (Kelleher et al. 2015). Some of these values do not tell us much information in this format, but for comparison purposes, we’ll put them in a table:

Feature	Mean	Median
ExternalRiskEstimate	71.800843	72.0
MSinceOldestTradeOpen	194.305585	177.0
MSinceMostRecentTradeOpen	9.374078	6.0
PercentTradesNeverDelq	91.970495	97.0
MaxDelq2PublicRecLast12M	5.727778	6.0
MaxDelqEver	6.322222	6.0
NumTradesOpeninLast12M	1.891465	1.0
MSinceMostRecentInqexcl7days	0.142255	0.0
NumInqLast6M	1.420443	1.0
NumInqLast6Mexcl7days	1.353003	1.0

AverageMInFile	77.778714	75.0
PercentInstallTrades	35.463646	33.0
NetFractionRevolvingBurden	34.987355	31.0
NetFractionInstallBurden	41.799789	50.0
NumRevolvingTradesWBalance	3.732350	3.0
NumInstallTradesWBalance	1.604847	2.0
NumBank2NatlTradesWHighUtilization	0.513172	1.0
NumSatisfactoryTrades	20.581665	19.0
NumTrades60Ever2DerogPubRec	0.679663	0.0
NumTrades90Ever2DerogPubRec	0.453109	0.0
PercentTradesWBalance	66.678609	67.0

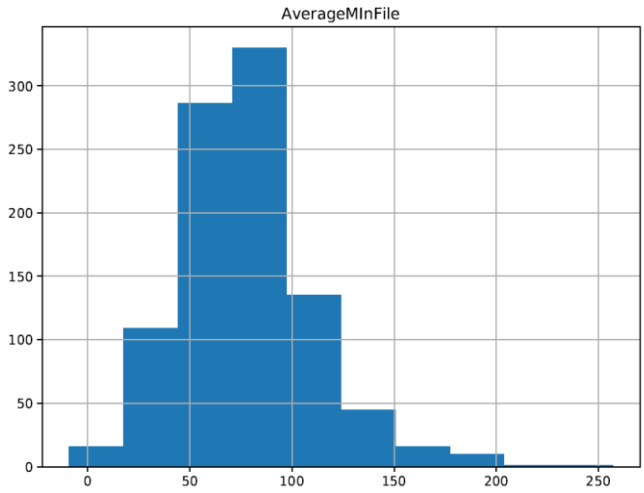
Based on the table above, it makes it a little easier to spot significant differentials. The features to pay attention to and also identify whether such differentials impact the plots carried out below are:

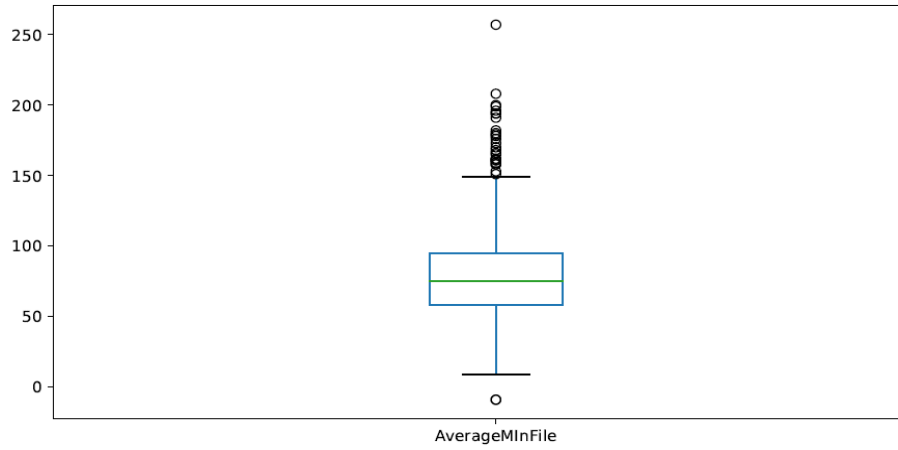
- PercentTradesNeverDelq
- NetFractionInstallBuden

*Note: the above are simply being treated as an **initial** flag. Other data quality issues may be identified in plots below.*

After plotting the continuous columns we can see that many of our plots follow a Gaussian curve, or are skewed left or right. This makes it clear to spot potential issues in the data, such as outliers (values which are skewing representation). In general, I was happy with the results of the histograms, though some data issues such as outliers, erroneous entries and logical errors may need to be addressed.

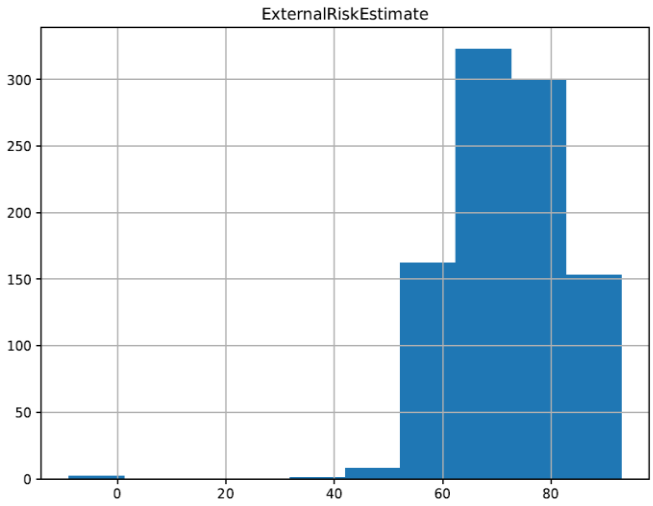
1.6.1 AverageMInFile

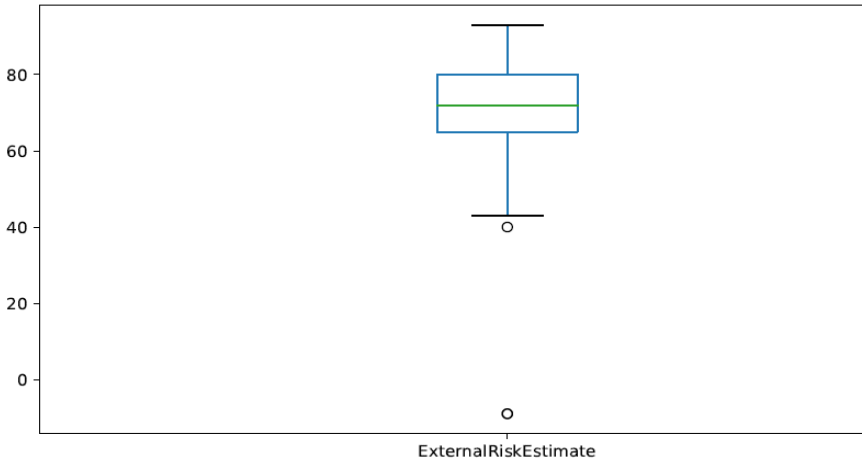




This plot is a unimodal skewed right, which appears to make sense. It does highlight a slowing of growth, with a small amount of months in the file at the older stages and also a smaller number of “contracts” more recently, while the bulk is concentrated around a central tendency. This would suggest slowing growth of new contracts within the data. The negative value should be dealt with.

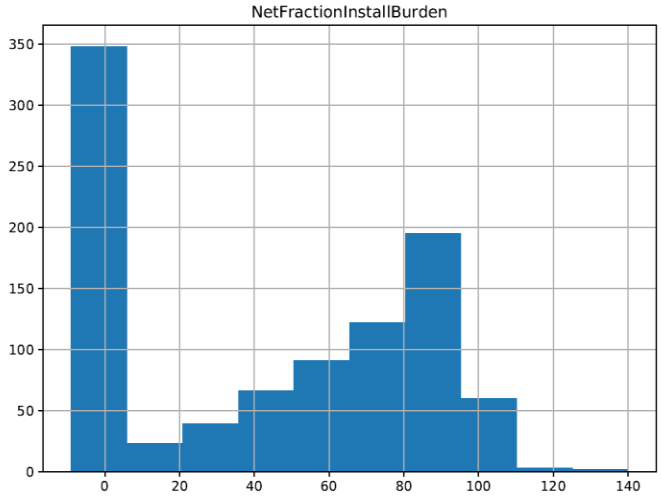
1.6.2 ExternalRiskEstimate

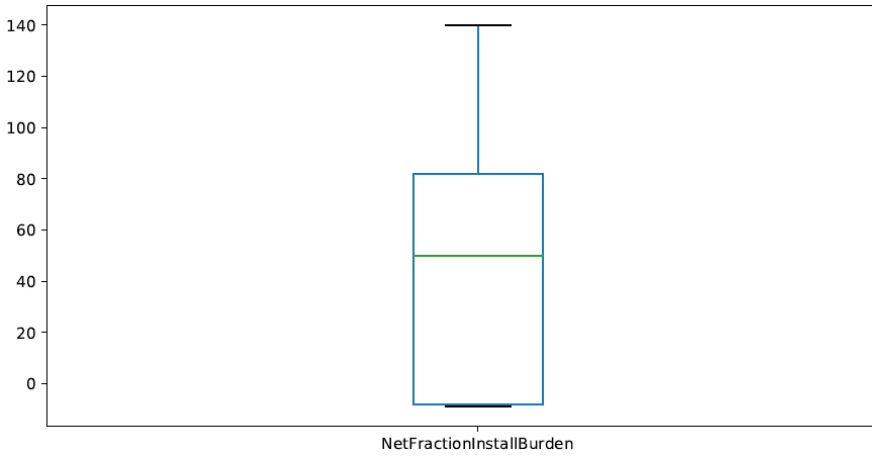




While following somewhat of a normal distribution. It might be worth redistributing the features bin size in order to further clarify the tendency. The curve looks a little suspicious since the graph is forced far out to the right of the graph. This is due to a negative value which doesn't make sense, since you cannot have "negative risk" in theory.

1.6.3 NetFractionInstallBurden

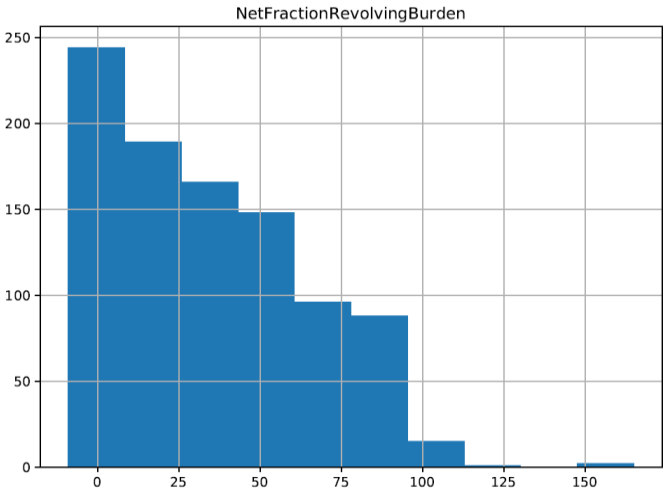


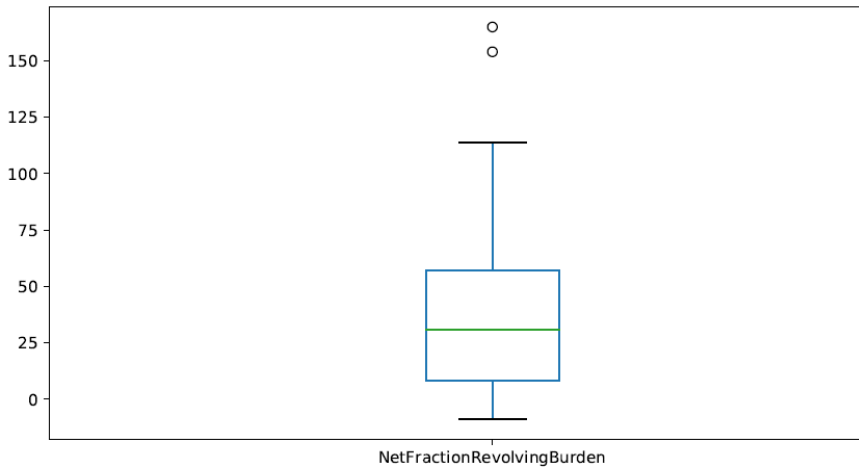


This histogram plot could represent a unimodal skewed left. The presence of negative values is skewing the interpretation of these plots. Upon further investigation, I identified this issue as the presence of many negative (-8) values. These are encoded with a special meaning. Though they are useful for the grand scheme of analysis, they do not make sense when combined with numerical analysis in this aspect. These -8 values should be dealt with in the data quality plan.

It should be noted that a high percentage of data would not be ideal to lose, so alternative avenues should be considered rather than removing the feature/instances.

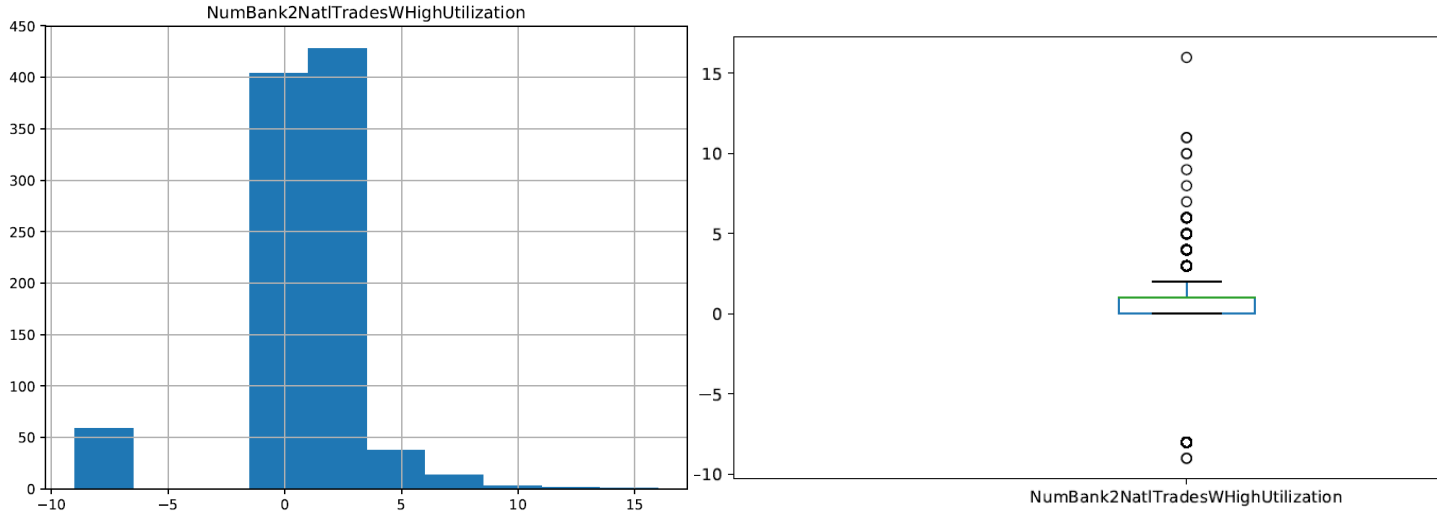
1.6.4 NetFractionRevolvingBurden





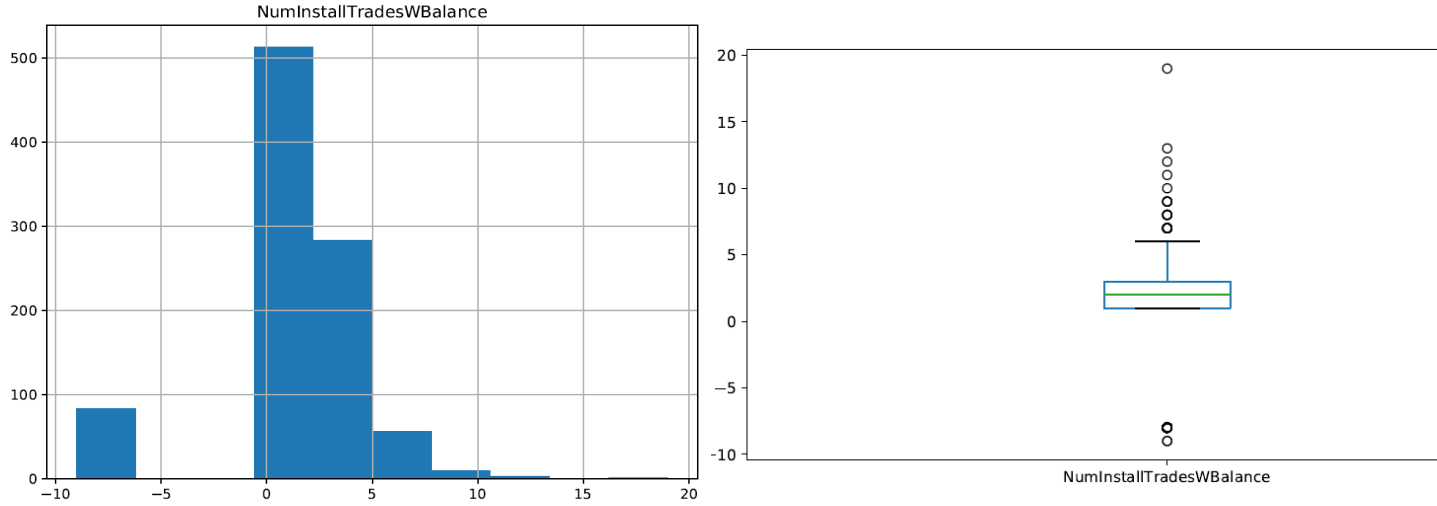
This plot is unimodal skewed right. The presence of negative values is suspicious. Upon further investigation, we notice as above that there are some -8 values with special meanings again coded into a numerical analysis. These should be dealt with in the data quality plan.

1.6.5 NumBank2NatlTradesWHighUtilization



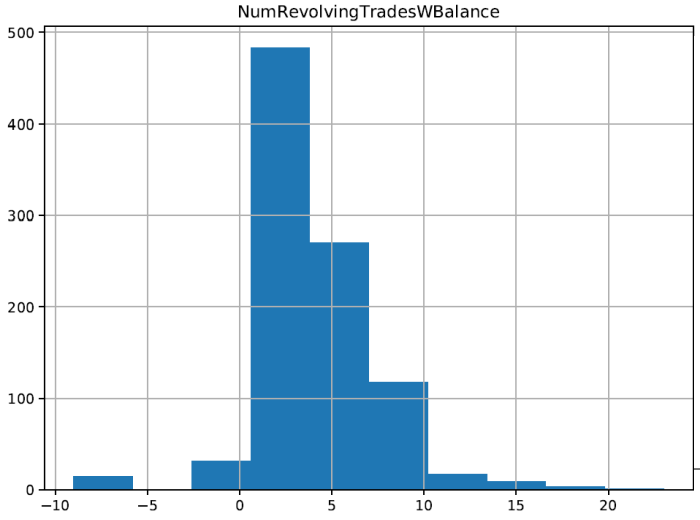
This plot is interesting. The view of the data is being drastically hindered by values around the central tendency. This could mean that the bins should be redefined in order to have a more appropriate representation. The negative values are once again a result of some -9 and -8 values in the dataset with special meanings. These should be removed for numerical analysis.

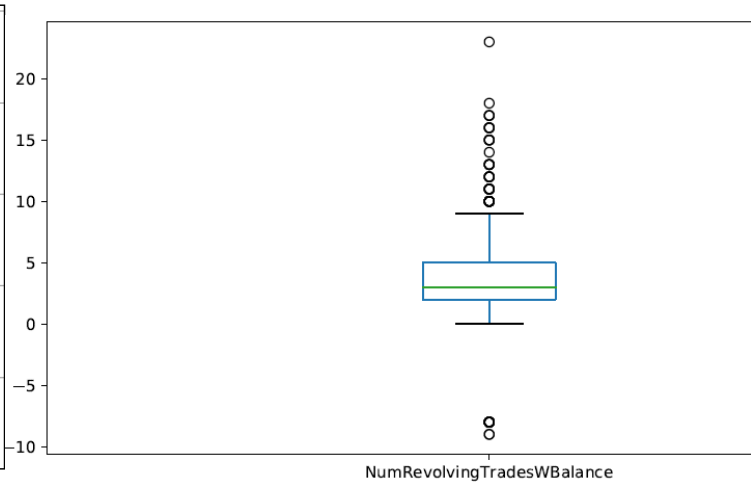
1.6.6 NumInstallTradesWBalance



Again we see many values concentrated around the central values. This could indicate that we might have a better visualization of our data if we redefine the bin size. Presence of negative values also indicates special meaning values are once again found in this features. Should be dealt with in the data quality plan.

1.6.7 NumRevolvingTradesWBalance

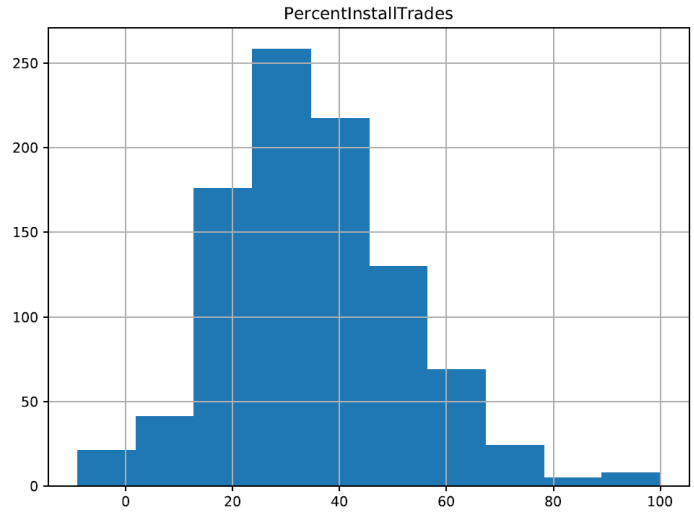


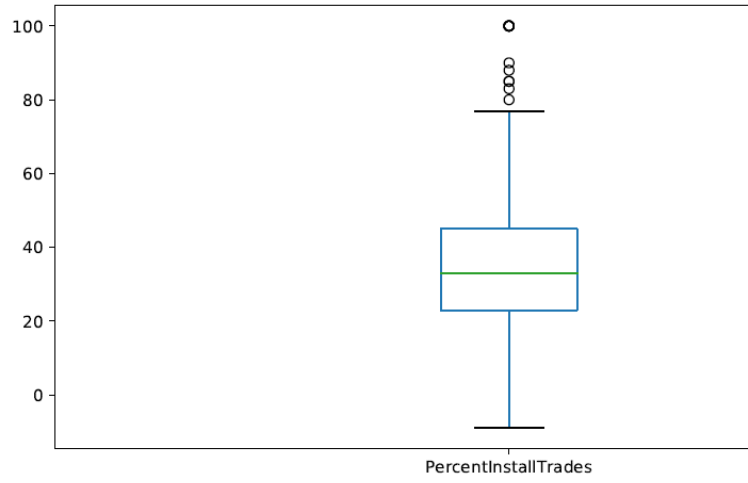


This plot is skewed right, but majority of the values are concentrated around the center between 0-10. This might again demonstrate that too many values are concentrated in the central bins. Reducing their size may give a more clarifying picture of the data.

Additionally, the box plot clearly shows us that we have presence of -9 and -8 values once again.

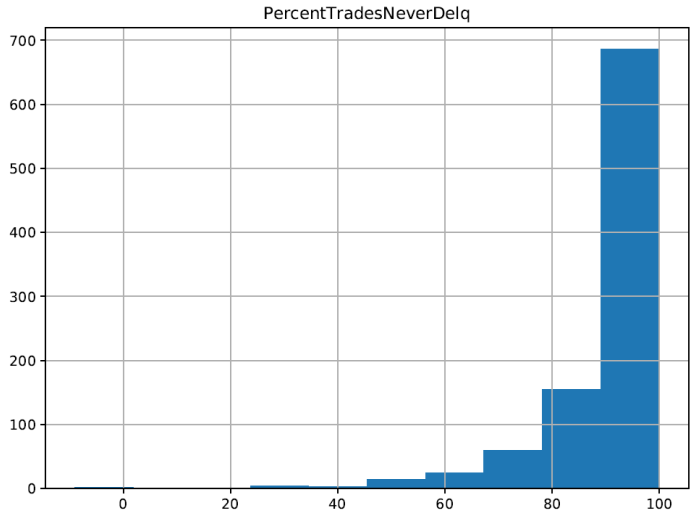
1.6.8 PercentInstallTrades

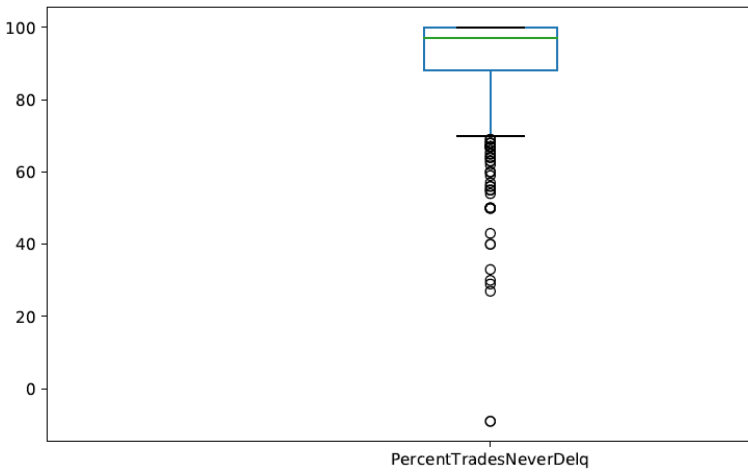




Percent install trades follows a normal distribution. Since the data looks spread well, this data may be okay and require no further modifications (except for dealing with the negative values).

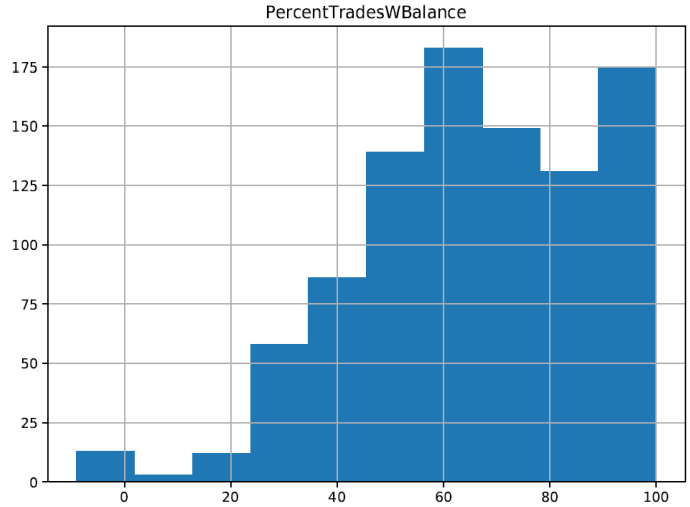
1.6.9 PercentTradesNeverDelq

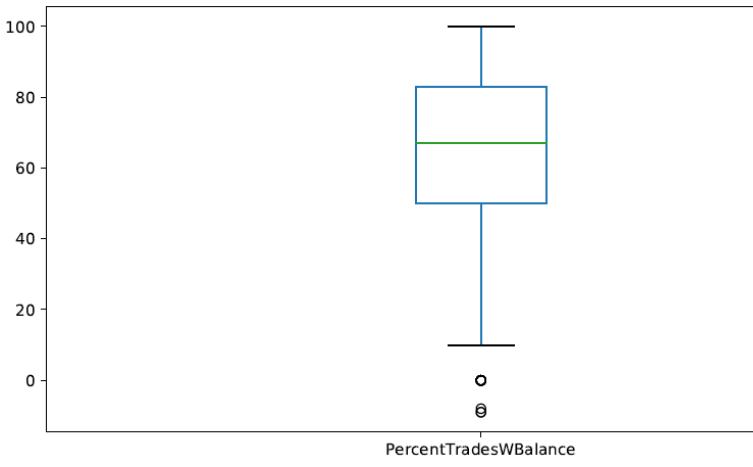




I can identify this graph grows exponentially, indicating that the percentage of trades that were never delinquent is quite high. This was suspicious, since other features did not indicate that this would be the case. Upon further investigate, I briefly checked this feature against the “RiskPerformance”. One would assume that if 100% of the account’s trades were “never delinquent”, this would imply that we should have a “good” risk performance rating. This was not the case. The data indicated that “good” and “bad” have both been allocated to accounts that have never been delinquent and as mentioned, this is suspicious. It’s worth investigating the percentage of never delinquent accounts with bad ratings, and if high, dealing with this issue in the data plan.

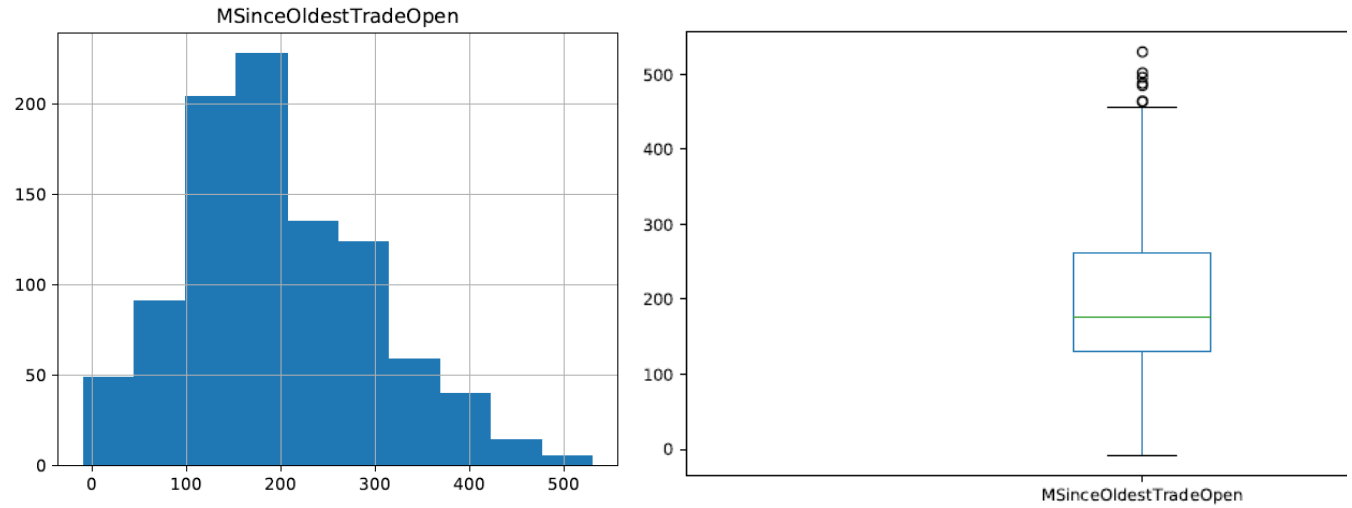
1.6.10 PercentTradesWBalance





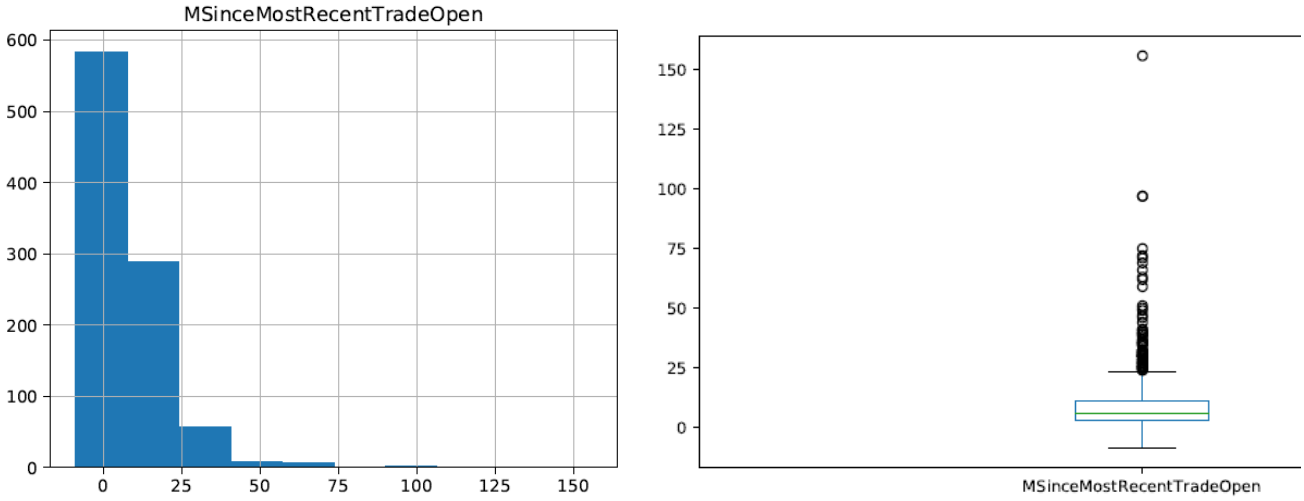
This graph identifies that it could follow a normal distribution, but a values between 90 - 100 mark prevent this from being a normal distribution. It might make sense to look into this value and possibly cap our values depending on further analysis. Clamping the thresholds might work here.

1.6.11 MSinceOldestTradeOpen



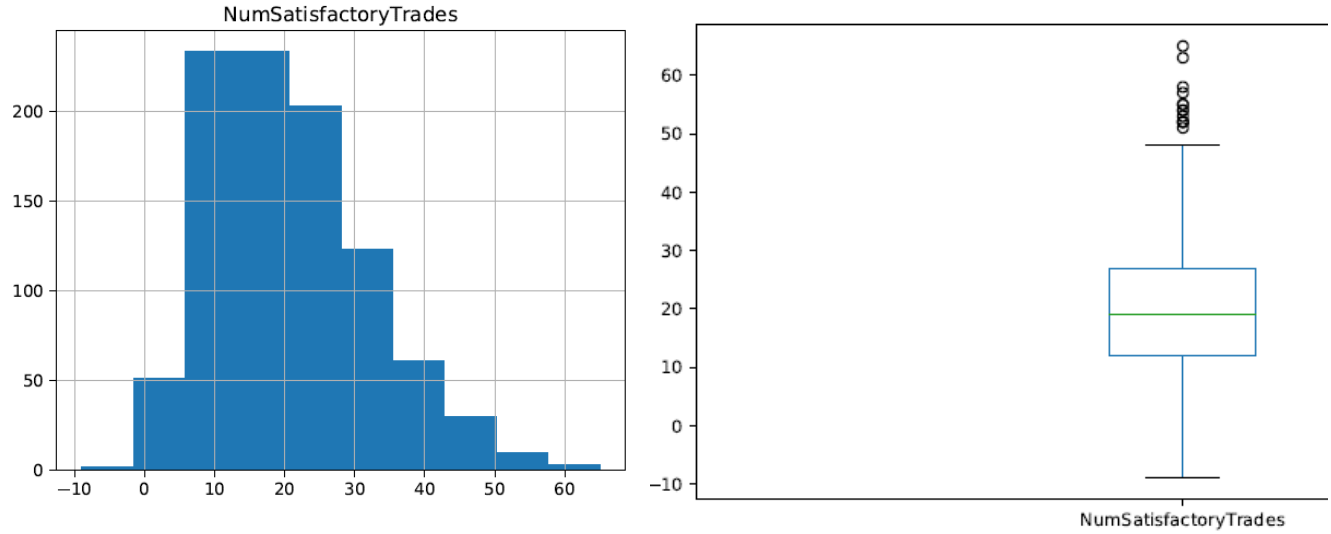
MSinceOldestTradeOpen visualisations seem like an accurate representation of the data when considering the context. Taking this as a measure of customer activity, it makes sense that only a small proportion of our subjects have trades open more than 400 months ago with that figure steadily rising the closer we get to the peak (mimicking business growth).

1.6.12 MSinceMostRecentTradeOpen



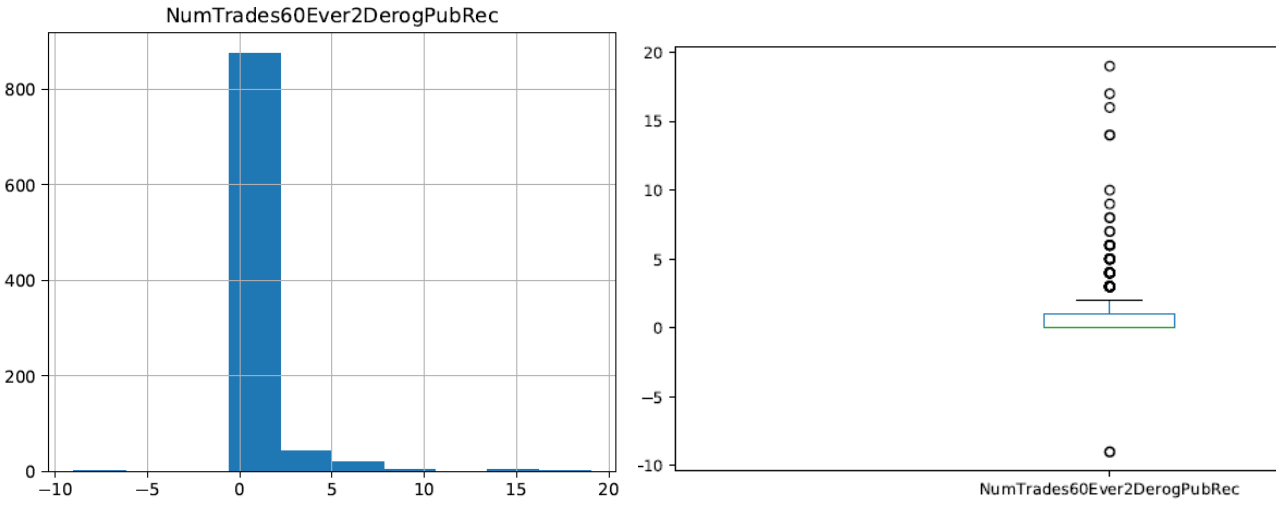
These plots also seem good for our analysis. They make it clear that the vast majority of subjects in the dataset have had a trade within the last 0 months, and the remaining majority having a trade within the last 25 months. The box plot further backs this up by having a small range. It shows the outliers in the data, but these outliers are not enough to skew the data and thus no further action is needed.

1.6.13 NumSatisfactoryTrades



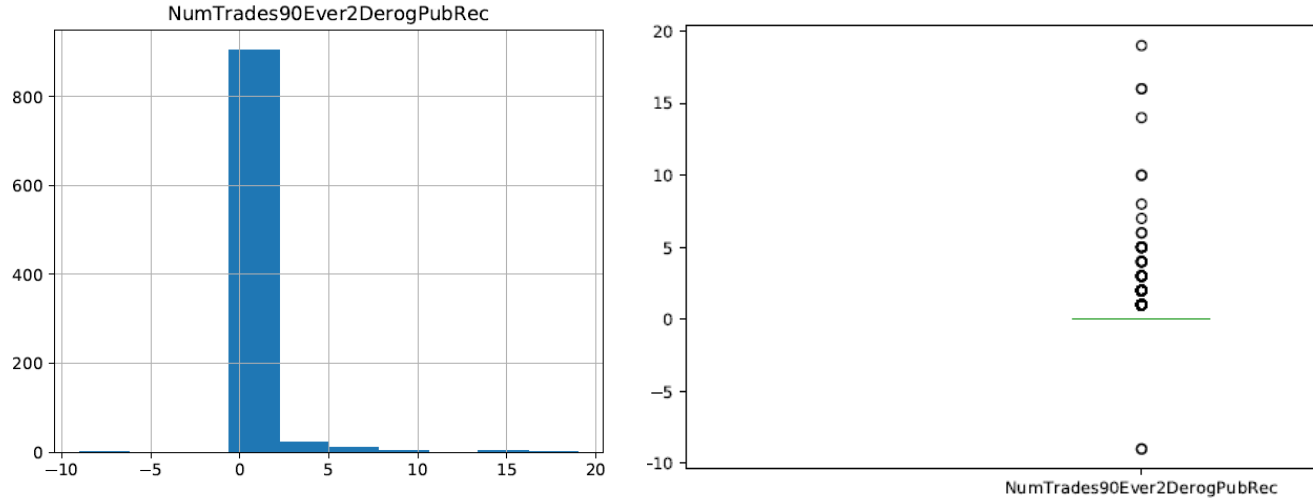
The number of satisfactory trades follows a skewed right distribution with a central tendency of 15 satisfactory trades. However, the presence of the negative value could represent something erroneous that should be dealt with in the data plan. This measure isn't supposed to account for the number of dissatisfactory trades and thus a negative value doesn't appear to make sense. It should start at 0, at least.

1.6.14 NumTrades60EverDerogPubRec



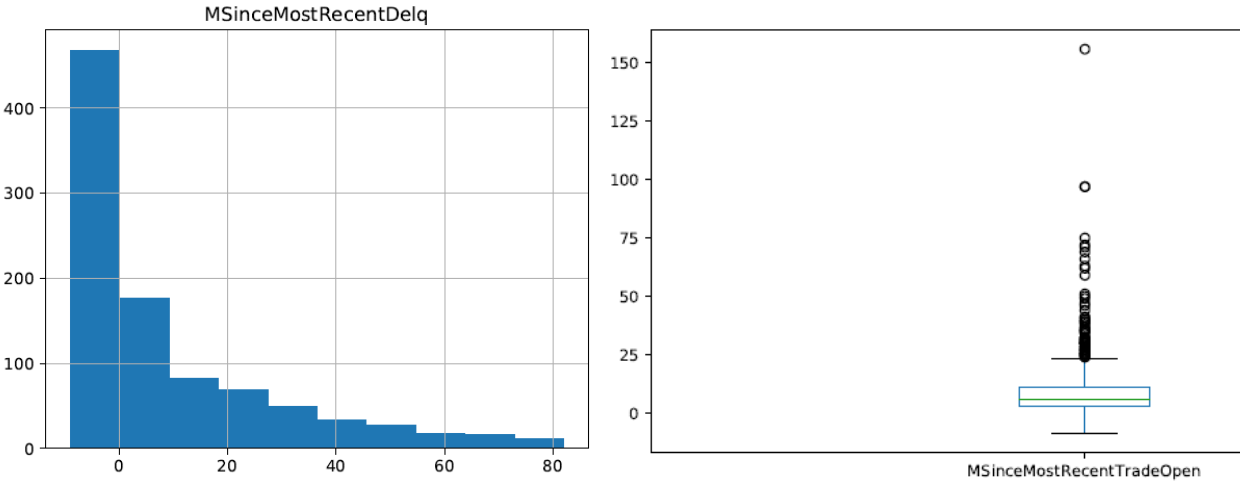
For these plots, again, it doesn't really make sense that we have negative values. This is accounting for the number of subjects which have over 60 trades ever. You cannot have minus trades. The box plot allows us to identify that there are very few accounts that have over 60 trades, and clearly shows us the rough value of those who do. It also more clearly identifies that there is only 1 culprit for the negative value and thus this should be removed.

1.6.15 NumTrades90EverDerogPubRec



As above in 1.6.14, except the range of values with over 90 trades is even smaller.

1.6.16 MSinceMostRecentDelq

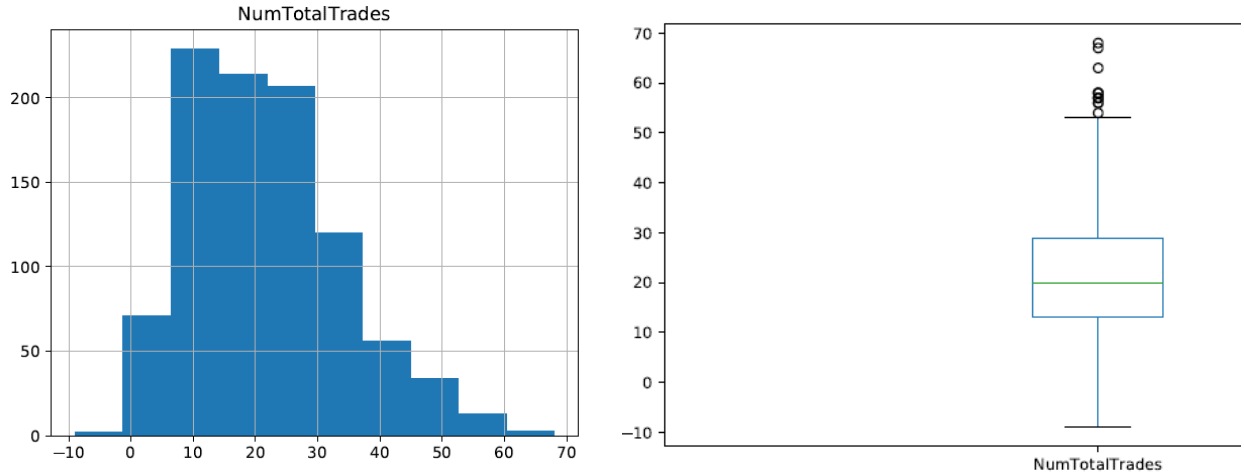


We can see that there are plenty of negative values. These negative values (-7 and -8 upon further inspection) carry a special meaning. -7 implies no delinquencies and –8 suggests that the data is not available. A count on this feature suggests that 46% of the data in this column is in fact no use for statistical analysis. It could be an option to convert this feature to categorical, which would allow us to ascribe meanings to the labels. Otherwise, we could consider dropping the feature if we deem it unnecessary in determining the value of the target feature. As per instruction in Kelleher (et. al 2015), values this high need to be remedied when they're missing, but in this case it's not necessarily missing it just isn't appropriate to include in this graph.

Another interesting scenario here. It presents the question whether or not you can have 0 months since the last delinquency. For example, it would depend on when the data is collected for each month. These analysis are generally retrospective. Therefore, it may not make sense that we have 0 months since the last delinquency (imagine filing results for the previous month, you might list it as "1" month since most recent delinquency). Therefore, this issue should be considered in the data quality plan. And since -7 in

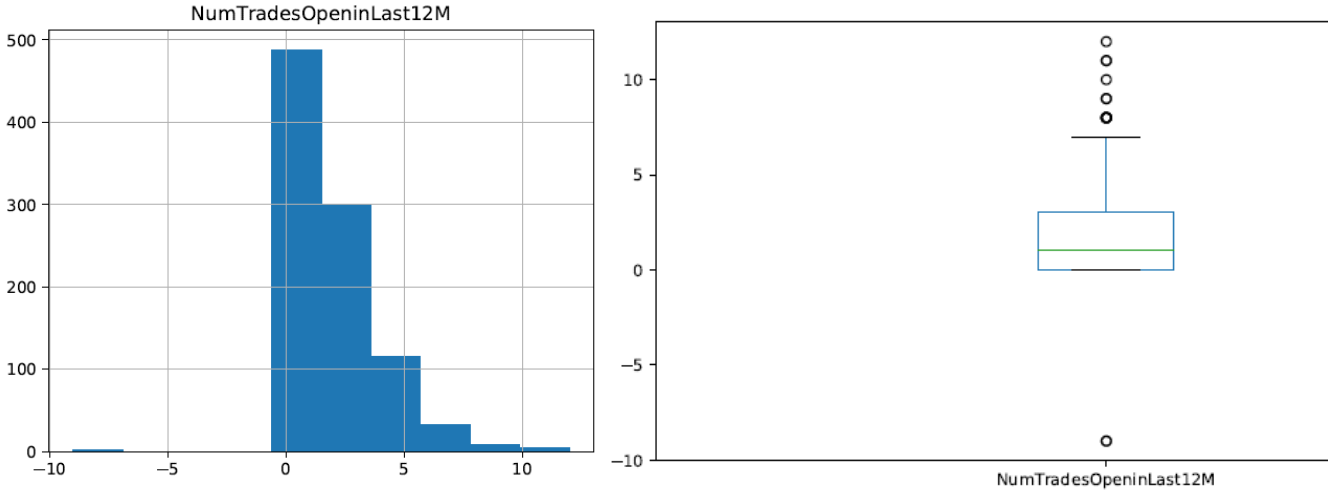
encoded with the meaning of “never had a delinquency”, then 0 is either erroneous or the data was collected in two different ways and 0 should represent “never had a delinquency” as well. This is another issue that should be considered in the data quality plan.

1.6.17 NumTotalTrades



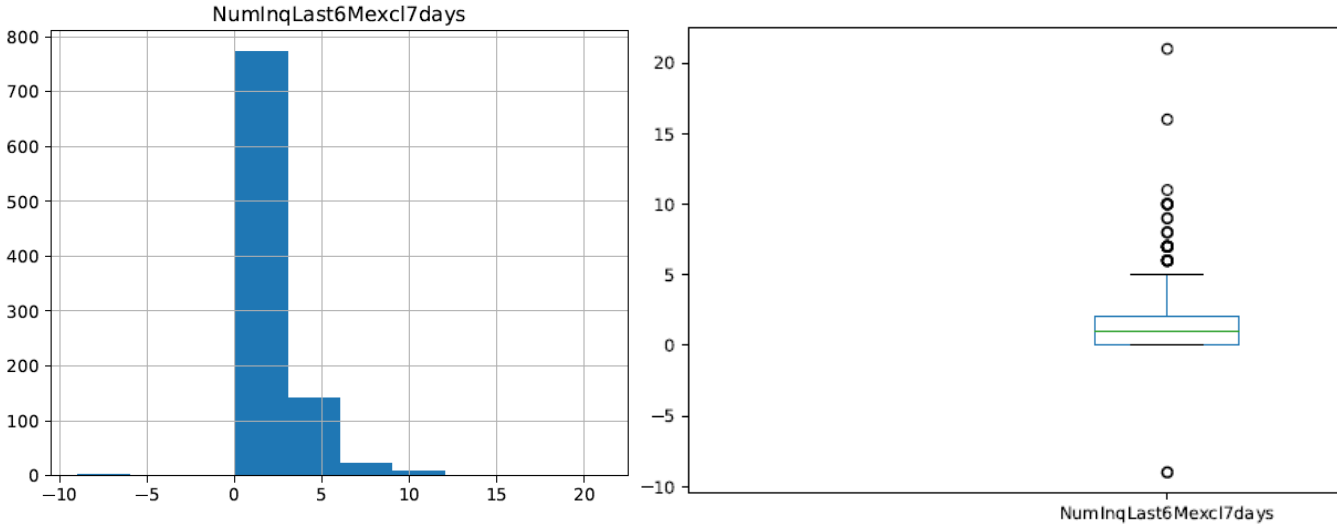
Again, the presence of negative values should be dealt with considering you cannot have a negative number of trades. Ignoring the negative value, we see that the plot is skewed right with a small number of accounts having a large number of trades.

1.6.18 NumTradesOpenInLast12M



Similarly, we should ignore the negative value here as we know we have to deal with this. Other than that, I can’t identify any major issues with this plot. It’s skewed right with many accounts having no trades in the last 12 months. Hypothetically, this should mean that most other data recorded for the previous 12 months should also be 0 for each subject (with the exception of inquiries).

1.6.19 NumInq6Mexcl7days



No identifiable issues with this feature other than the negative value.

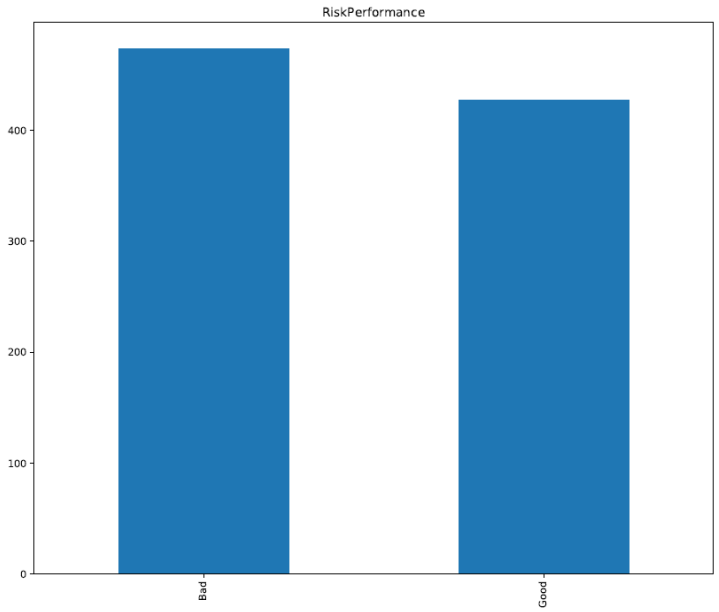
1.7 Categorical Features

1.7.0 Statistical Overview

	count	unique	top	freq
RiskPerformance	900	2	Bad	473
MaxDelq2PublicRecLast12M	900	10	7	382
MaxDelqEver	900	8	8	408

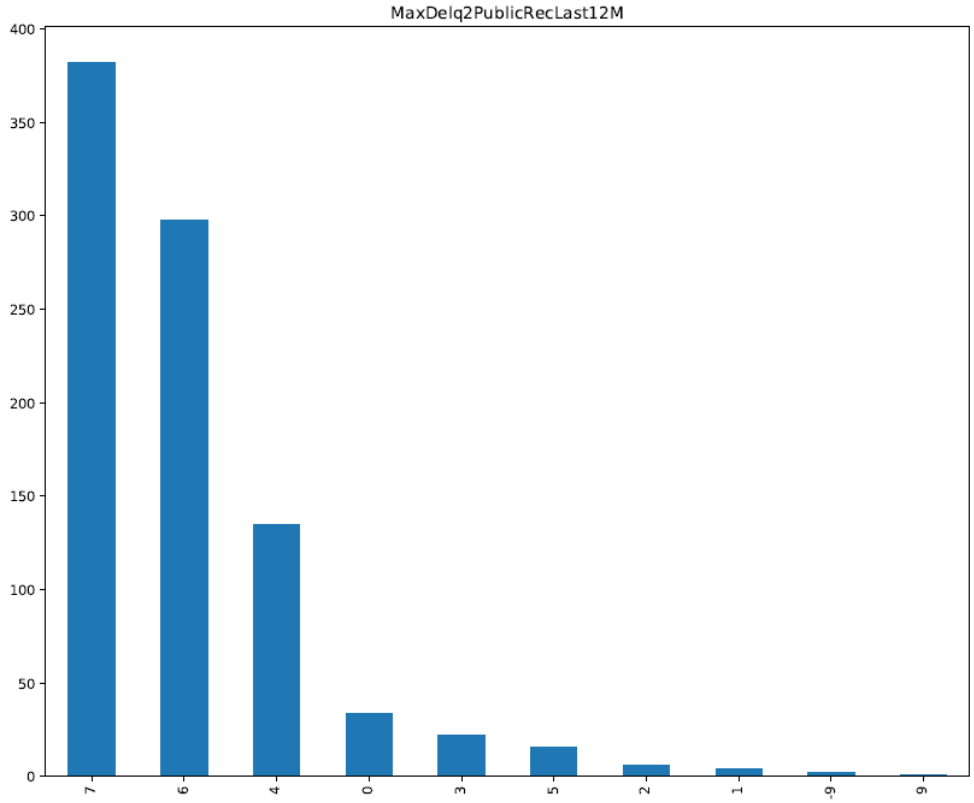
The data above displays only 900/949 as the remaining 49 values are null. These values have been excluded from the plots below.

1.7.1 RiskPerformance



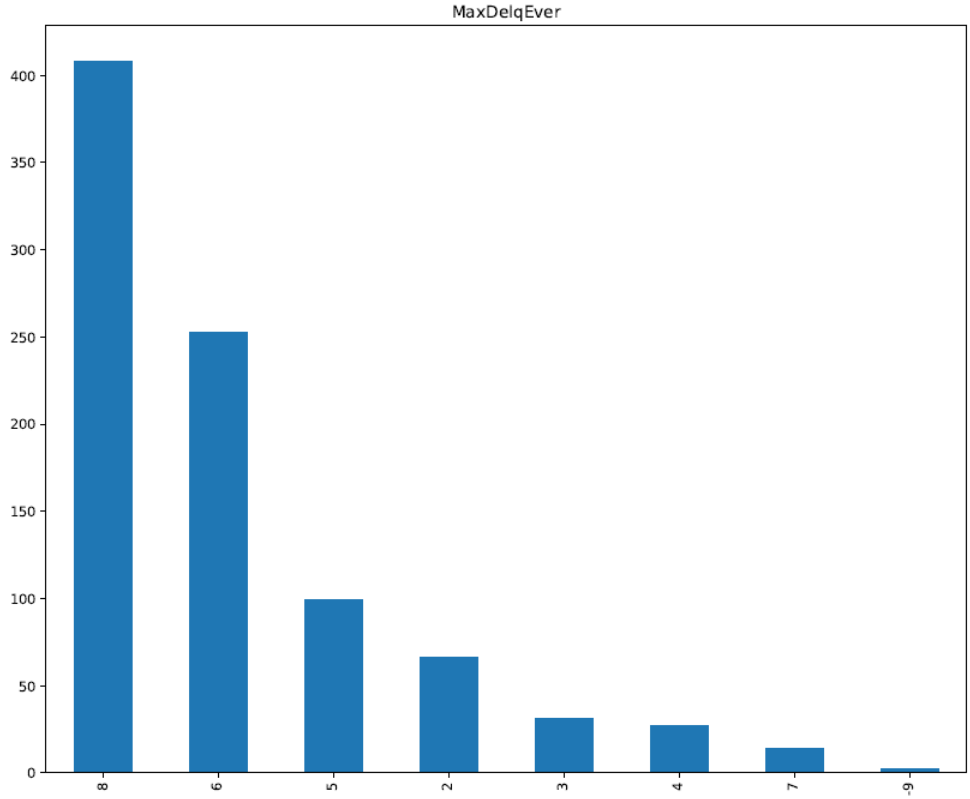
Data appears to be balanced on two binary values. It clarifies that roughly half of our dataset has been graded as bad, while the remaining half graded as good. We are content with this as it should mean our data is roughly representative and our sample may give us an accurate depiction of what causes good versus bad RiskPerformance ratings (which is our target feature).

1.7.2 MaxDelq2PublicRecLast12M



We can see that this feature is heavily skewed right. However, upon further inspection, values 5 and 6 represent “unknown delinquency”. Assessing the weighting of all values of 5 and 6 is necessary to determine whether we should drop the feature or not. Also, the negative values should be removed.

1.7.3 MaxDelqEver



This feature paints a slightly more reasonable picture. The only issue is removing the negative 9. All other results are coded in the data dictionary, and the results seem reasonable. No further action is required.